

03

CSS3 là gì?



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



CSS3 khác gì với CSS

- CSS3 là một chuẩn (phiên bản) mới nhất của CSS.
- CSS3 tạo ra flexbox thay thế cho float.
- CSS3 tạo ra hệ thống lưới (grid) phục vụ dàn layout
- CSS3 cung cấp các thuộc tính mới làm những điều CSS không thể làm.
- CSS3 có thể tạo được các animation.

CYBERLEARN
ĐÀO TẠO CHUYÊN NGHIỆP TRÌNH



Các thuộc tính của CSS3 được sử dụng:

- Border-radius
- Background-size
- Linear-gradient
- Text-shadow
- Box-shadow
- Before,after
- @font-face
- CSS variables

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

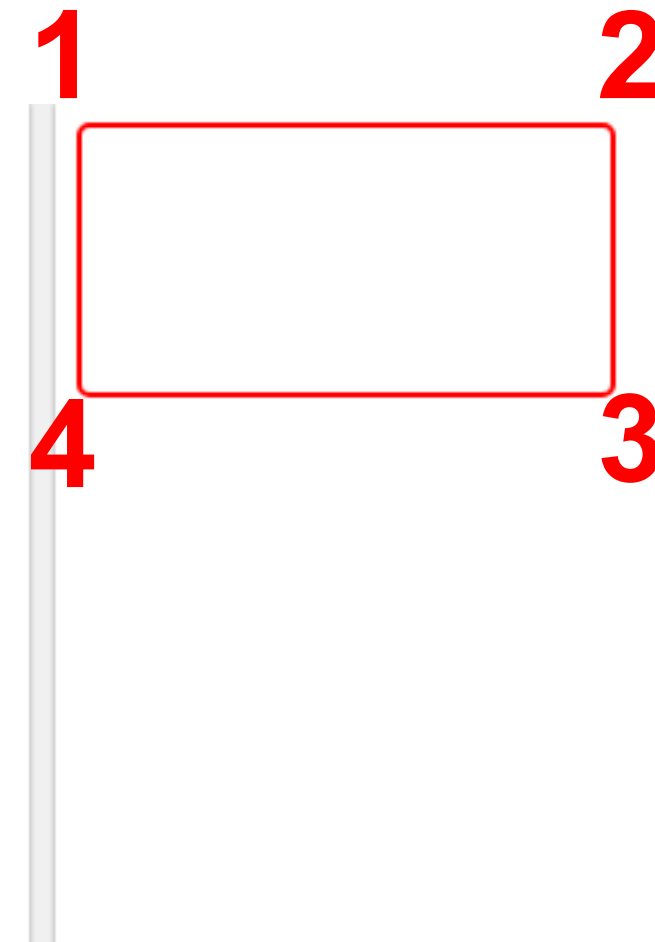
Các thuộc tính của CSS3

- Border-radius:

- Thuộc tính border-radius : bo tròn góc các phần tử
- Có thể theo px , rem, em hoặc %;
- Cú pháp và kết quả :



```
<!DOCTYPE html>
<html>
<head>
<style>
div{
  width:200px;
  height:100px;
  border: 2px solid red;
  border-radius:10px;
}
</style>
</head>
<body>
  <div></div>
</body>
</html>
```



- Các cách viết khác của border-radius

- 2 tham số : **border-radius : [góc 1,3] [góc 2,4]**
- 3 tham số : **border-radius : [góc 1] [góc 2,4] [góc 3]**
- 4 tham số : **border-radius : [góc 1] [góc 2] [góc 3] [góc 4]**

Các thuộc tính của CSS3

- background-size :
 - Theo px ,rem hoặc em
 - Theo %
 - Contain, cover ,auto

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
    border: 1px solid black;
    background:url(img_flwr.gif);
    background-size: 100px 80px;
    background-repeat: no-repeat;
    padding:15px;
}

#example2 {
    border: 1px solid black;
    background:url(img_flwr.gif);
    background-repeat: no-repeat;
    padding:15px;
}
</style>
</head>
<body>
```

Điều chỉnh kích thước background:



Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Mặc định



Lorem Ipsum Dolor

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Các thuộc tính của CSS3

- background-size :
 - Phân biệt cover và contain
 - Contain: Hình nằm trong content , tùy theo kích thước khung chứa mà nó sẽ full theo chiều rộng hoặc chiều cao , không bao phủ hết.
 - Cover : Hình bao phủ hết nền, tuy nhiên hình ảnh có thể bị scale để bao phủ hoặc nếu quá lớn thì sẽ mất đi một số phần.
 - Chúng ta có thể kết hợp với background- position để lấy vị trí hình

background-size: contain:



background-size: cover:



Các thuộc tính của CSS3

- Color : RGBA
 - Giá trị màu RGBA là một phần mở rộng của các giá trị màu RGB với 1 kênh alpha – xác định độ trong suốt của màu



```
<!DOCTYPE html>
<html>
<head>
<style>
#p1 {background-color:rgba(255,0,0,0.3);}
#p2 {background-color:rgba(0,255,0,0.3);}
#p3 {background-color:rgba(0,0,255,0.3);}
#p4 {background-color:rgba(192,192,192,0.3);}
#p5 {background-color:rgba(255,255,0,0.3);}
#p6 {background-color:rgba(255,0,255,0.3);}
</style>
</head>
<body>

<h1>Define Colors With RGBA Values</h1>

<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Grey</p>
<p id="p5">Yellow</p>
<p id="p6">Cerise</p>
```

Define Colors With RGBA Values



Các thuộc tính của CSS3

- Linear Gradients
 - CSS3 hỗ trợ 2 loại màu Gradients :
 - Linear Gradients : trải màu theo dưới/trên/trái/phải/chéo
 - Radial Gradients: trải từ tâm ra.
 - Cú pháp: background: linear-gradient(direction, color-stop1, color-stop2, ...);

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background: red; /* For browsers that do not support gradients */
  background: -webkit-linear-gradient(red, yellow);
  background: -moz-linear-gradient(red, yellow);
  background: -o-linear-gradient(red, yellow);
  background: linear-gradient(red, yellow); /* Standard syntax (must be
last) */
}
</style>
</head>
<body>

<h1>Linear Gradient - Top to Bottom</h1>
<p>This linear gradient starts at the top. It starts red, transitioning to
```

Linear Gradient - Top to Bottom

This linear gradient starts at the top. It starts red, transitioning to



Note: Internet Explorer 9 and earlier versions do not support

Linear Gradient : Trải từ Top đến bottom (Mặc định)

Các thuộc tính của CSS3

- Linear Gradients

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
    height: 200px;
    background: red; /* For browsers that do not support gradients */
    background: linear-gradient(to bottom right, red, yellow); /* Standard
syntax (must be last) */
}
</style>
</head>
<body>

<h1>Linear Gradient - Diagonal</h1>
<p>This linear gradient starts at top left. It starts red, transitioning to
yellow:</p>

<div id="grad1"></div>
```

Linear Gradient - Diag

This linear gradient starts at top left. It starts red,



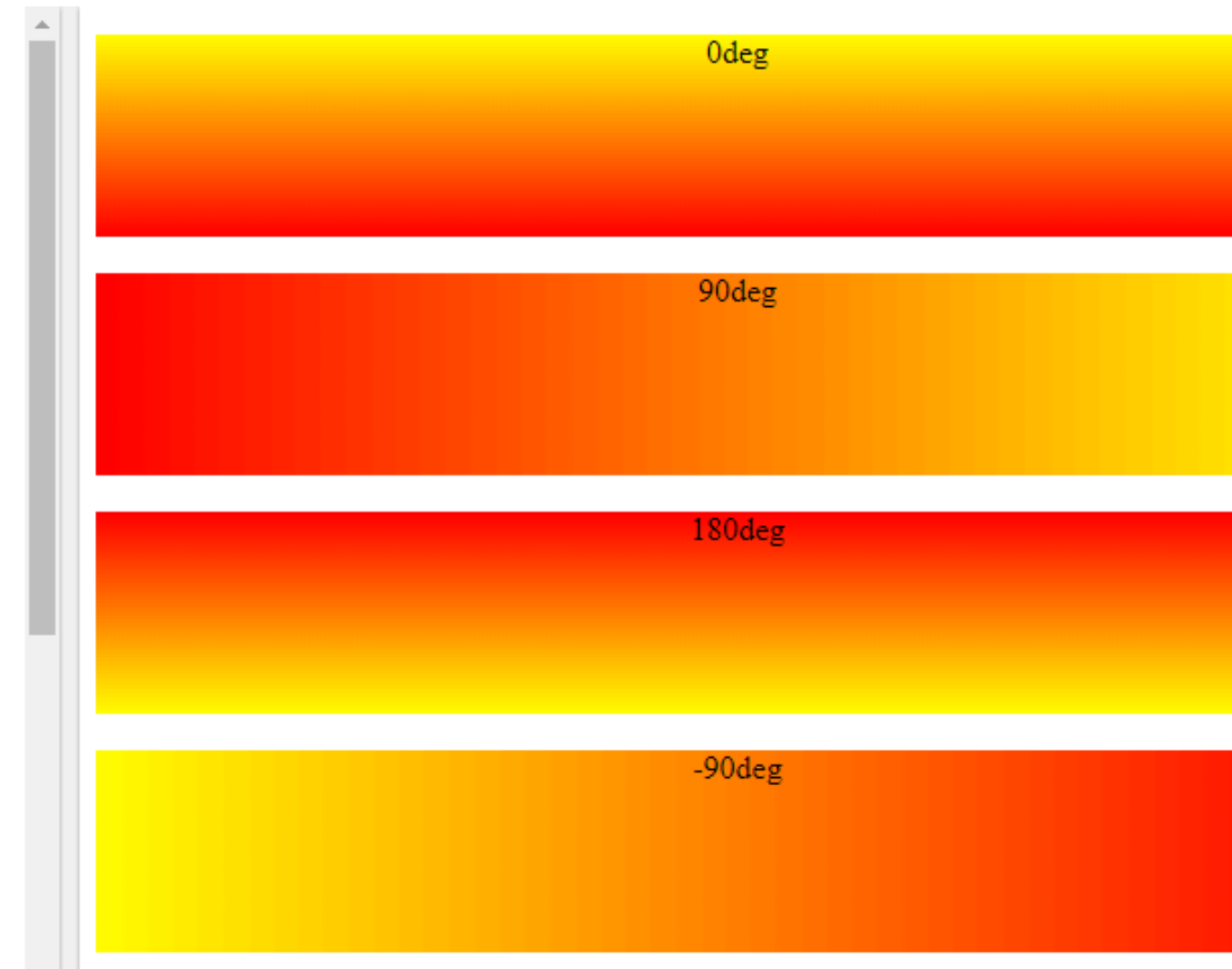
Note: Internet Explorer 9 and earlier versions do

Linear Gradient: Trái theo đường chéo

Các thuộc tính của CSS3

- Linear Gradients

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 100px;
  background: red; /* For browsers that do not support gradients */
  background: linear-gradient(0deg, red, yellow);
}
#grad2 {
  height: 100px;
  background: red; /* For browsers that do not support gradients */
  background: linear-gradient(90deg, red, yellow);
}
#grad3 {
  height: 100px;
  background: red; /* For browsers that do not support gradients */
  background: linear-gradient(180deg, red, yellow);
}
#grad4 {
  height: 100px;
  background: red; /* For browsers that do not support gradients */
  background: linear-gradient(-90deg, red, yellow); /* Standard syntax
(must be last) */
|
```

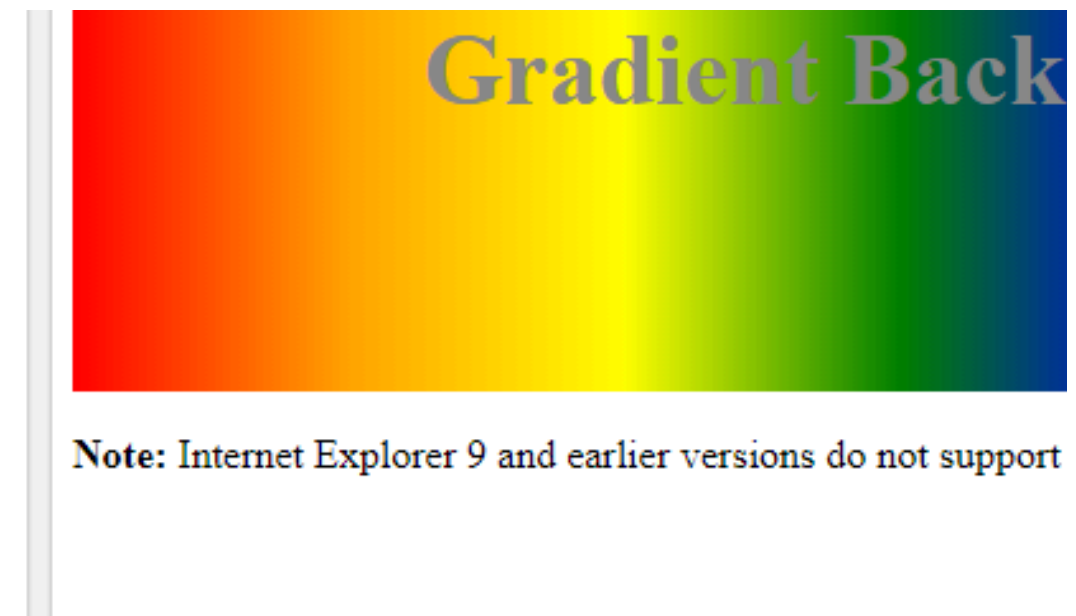


Linear Gradient : sử dụng
degree

Các thuộc tính của CSS3

- Linear Gradients

```
<!DOCTYPE html> | Change Orientation |
<html>
<head>
<style>
#grad1 {
  height: 150px;
  background: red; /* For browsers that do not support gradients */
  background: linear-gradient(to right, red, orange, yellow, green, blue,
indigo, violet); /* Standard syntax (must be last) */
}
</style>
</head>
<body>
```



Linear Gradient : kết hợp nhiều màu
trải

Các thuộc tính của CSS3

- Text Shadow
 - Vertical + Horizontal Shadow

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px;
}
</style>
</head>
<body>
```

Text-shadow effect

Note: Internet Explorer 9 and earlier

- Vertical + Horizontal + color

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px red;
}
</style>
</head>
<body>
```

Text-shadow effect!

Note: Internet Explorer 9 and earlier versions, do not support the

```
<h1>Text-shadow effect!</h1>
```

```
<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the
text-shadow property.</p>
```

Các thuộc tính của CSS3

- Text Shadow
 - Vertical + Horizontal + blur + color Shadow

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px 5px red;
}
</style>
</head>
<body>

<h1>Text-shadow effect!</h1>

<p><b>Note:</b> Internet Explorer 9 and earlier versions, do not support the
```

Text-shadow effect!

Note: Internet Explorer 9 and earlier versions, do not support the





Các thuộc tính của CSS3

- Box Shadow
 - Thuộc tính giúp làm nổi một khối các phần tử
 - Box-shadow:[val1] [val2] [val3] [val4] [color];

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 300px;
  height: 100px;
  padding: 15px;
  background-color: yellow;
  box-shadow: 10px 10px 10px 5px grey;
}
...

```

This is a div element with a box-shadow

Property				
box-shadow	10.0 4.0 -webkit-	9.0	4.0 3.5 -moz-	5.1 3.1 -webkit-

Các thuộc tính của CSS3

- @font-face
 - Cài đặt font chữ vào file từ CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
@font-face {
    font-family: myFirstFont;
    src: url(sansation_light.woff);
}

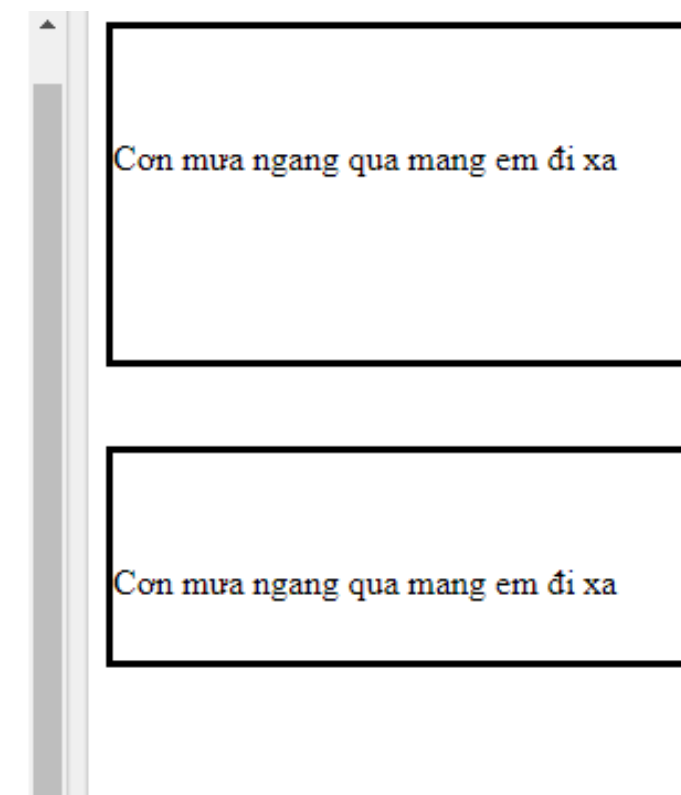
* {
    font-family: myFirstFont;
}
</style>
```

Change Orientation

Các thuộc tính của CSS3

- Resizing (box-sizing)
 - Cho phép thêm kích thước của border và padding vào width và height của 1 element (phần tử) hay không.
 - Cách tính khi không có box-sizing :
 - width+padding+border = chiều rộng của phần tử
 - height+padding+border = chiều cao của phần tử
 - Cách tính khi có box-sizing:
Vẫn lấy width height xét kích thước, nhưng nếu có padding thì không cộng thêm padding vào.

```
<html>
<head>
<style>
  #example1{
    width:300px;
    height:100px;
    border:3px solid black;
    padding-top:50px;
  }
  #example2{
    width:300px;
    height:100px;
    border:3px solid black;
    padding-top:50px;
    box-sizing:border-box;
  }
</style>
</head>
<body>
```



Các thuộc tính của CSS3

- Before và After
 - Cho phép chèn nội dung vào trước hoặc sau phần tử HTML.
 - Thường được dùng để tạo các symbol font (font awesome, material)

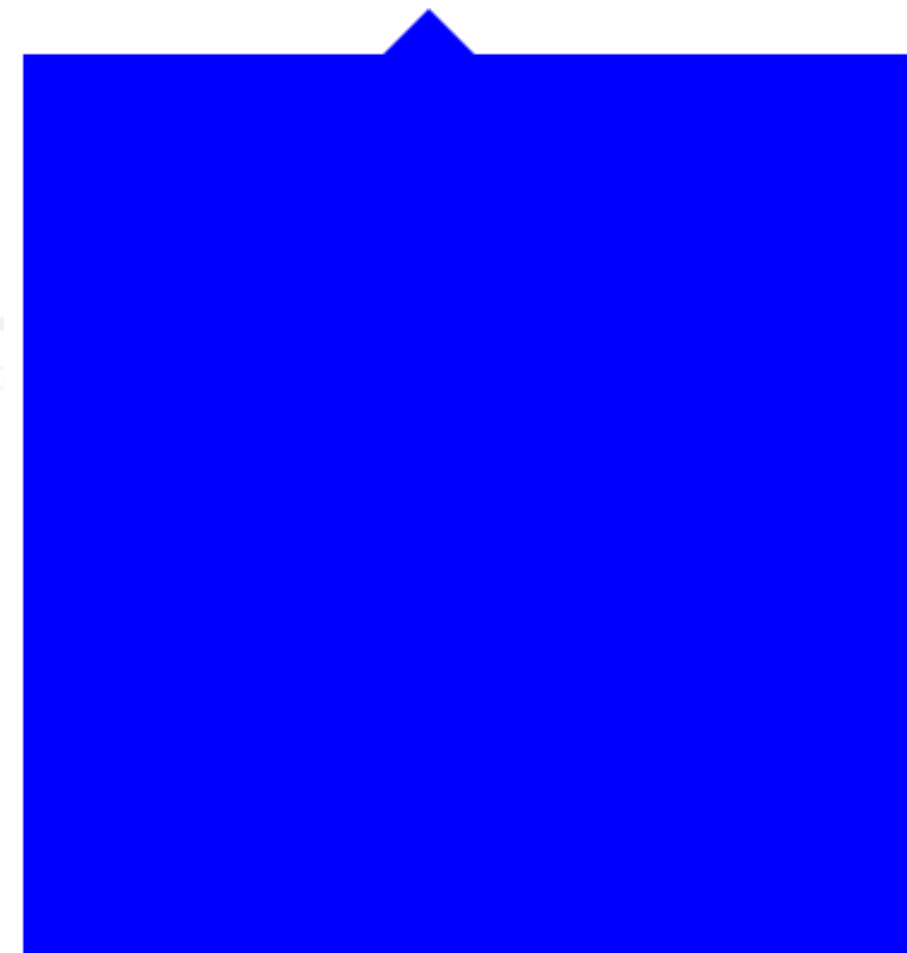


Các thuộc tính của CSS3

- Before và After
 - Sử dụng before hoặc after để tạo mũi tên

```
<style type="text/css">
  .demo{
    background-color:blue;
    width:300px;
    height:300px;
    position:relative;
    top:30px;
  }
  .demo::before{
    content:'';
    border-bottom:15px solid blue;
    border-left:15px solid transparent;
    border-right:15px solid transparent;
    position:absolute;
    top:-15px;
    left:40%;
  }
</style>
<body>

  <div class="demo"></div>
</body>
```



Các thuộc tính của CSS3

- Variable
- Tương tự như các ngôn ngữ lập trình khác, ta đặt và sử dụng biến thay thế cho giá trị của thuộc tính
- Tiết kiệm thời gian chỉnh sửa css

```
<style type="text/css">

:root{
  --color-red:#f00;
  --font-size:60px;
}
h1{
  color:var(--color-red);
  font-size:var(--font-size);
}

</style>
<body>

  <h1>CyberSoft Academy</h1>

</body>
</html>
```

Khu vực khai báo biến:
--Tên biến : giá trị

Cách sử dụng biến :
var(--Tên biến)

CyberSoft Academy



CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CSS3 Transition

CSS3 Transition

- Transition
 - Giúp chúng ta nhìn thấy được quá trình chuyển đổi của css
 - Để thực hiện transition cần có:
 - Thuộc tính CSS chúng ta cần thực hiện transition
 - Duration của hiệu ứng.
 - Ngoài ra chúng ta có các thuộc tính khác (có thể có hoặc không)
 - Timing-function
 - delay

CSS3 Transition

- Transition property
 - Chỉ định thuộc tính css chúng ta muốn thực hiện transition
 - Cú pháp:
 - Transition-property: [Tên thuộc tính] / all ;
 - Ví dụ :
 - Transition-property: width;
 - Transition-property: all;

CSS3 Transition

- Transition Delay

- Khoảng thời gian delay trước khi thực hiện hiệu ứng.

- Cú pháp:

- Transition-delay: [Time] ;

- Ví dụ : Transition-delay: 1s;

- Transition duration

- Thời gian thực hiện transition.

- Cú pháp:

- Transition-duration: [time];

- Ví dụ : Transition-duration : 2s;

CSS3 Transition

- Transition timing function
 - Thuộc tính transition-timing-function xác định tốc độ của hiệu ứng
 - Bao gồm các giá trị:
 - Ease – Hiệu ứng ban đầu chậm, sau đó nhanh và chậm dần
 - Linear – Tốc độ giữ nguyên từ đầu tới cuối
 - Ease-in – Bắt đầu chậm, sau đó nhanh
 - Ease-out – Bắt đầu nhanh, sau đó chậm
 - Ease-in-out – Bắt đầu và kết thúc chậm

CSS3 Transition

- Transition – cách viết shorthand và đầy đủ

```
.demo1,{  
    transition-property: width;  
    transition-duration: 1s;  
    transition-timing-function: linear;  
    transition-delay: 1s;  
}  
.demo2{  
    width: 200px;  
    height: 200px;  
    background-color: red;  
    transition:all 2s linear 1s;  
}
```

CSS3 Transition

- Ví dụ: transition giúp thấy rõ sự thay đổi kích thước của thẻ div khi hover

```
.demo1{  
    width: 200px;  
    height: 200px;  
    background-color: red;  
    transition:all 1s linear 0s;  
}  
.demo1:under{  
    width: 100px;  
}
```








CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

CSS3 Transform

CSS3 Transform

- CSS3 cung cấp cho ta một số thuộc tính để tạo hiệu ứng, làm cho trang web thêm sinh động
- Cho phép di chuyển, xoay , phóng to, thu nhỏ, làm nghiêng các phần tử trong trang web.
- CSS3 hỗ trợ transform 2D và 3D
- Trình duyệt hỗ trợ

Property					
transform	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	9.0 3.2 -webkit-	15.
transform-origin (two-value syntax)	36.0 4.0 -webkit-	10.0 9.0 -ms-	16.0 3.5 -moz-	9.0 3.2 -webkit-	15.



CSS3 Transform

- Các hàm sử dụng kèm với transform
 - Translate()
 - Rotate()
 - Scale()
 - SkewX()
 - SkewY()
 - Matrix()

CYBERLEARN
ĐÀO TẠO CHUYÊN NGHIỆP RÌNH

CSS3 Transform

- Translate

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <link rel="stylesheet" type="text/css" href="style.css"
7     />
8 </head>
9 <body>
10     <div class="demo"></div>
11 </body>
12 </html>
```

```
1 .demo{
2     width: 300px;
3     height: 200px;
4     background-color: red;
5 }
6 .demo{
7     -ms-transform: translate(120px,100px);
8     -webkit-transform: translate(120px,100px);
9     -moz-transform: translate(120px,100px);
10    -o-transform: translate(120px,100px);
11    transform: translate(120px,100px);
12 }
13
14 Nếu muốn di chuyển theo một chiều(Chỉ d
15
16 .demo{
17     transform: translateX(120px);
18 }
19
20 hoặc
21
22 .demo{
23     transform: translateY(120px);
24 }
```

CSS3 Transform

- Rotate

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <link rel="stylesheet" type="text/css" href="style.css"
   />
7 </head>
8 <body>
9   <div class="demo1"></div>
10  <div class="demo2"></div>
11 </body>
12 </html>
```

```
1 .demo1,.demo2{
2   width: 300px;
3   height: 200px;
4   background-color: red;
5   margin: 100px;
6   float: left;
7 }
8 .demo1{
9   -ms-transform: rotate(40deg);
10  -webkit-transform: rotate(40deg);
11  -moz-transform: rotate(40deg);
12  -o-transform: rotate(40deg);
13  transform: rotate(40deg);
14 }
15 .demo2{
16   -ms-transform: rotate(-40deg);
17   -webkit-transform: rotate(-40deg);
18   -moz-transform: rotate(-40deg);
19   -o-transform: rotate(-40deg);
20   transform: rotate(-40deg);
21 }
```

CSS3 Transform

- Scale

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6     <link rel="stylesheet" type="text/css" href="style.css"
7     />
8 </head>
9 <body>
10     <div class="demo1"></div>
11 </body>
12 </html>
```

```
1 .demo1,.demo2{
2     width: 300px;
3     height: 200px;
4     background-color:
5     margin: 100px;
6     float: left;
7 }
8 .demo1{
9     -ms-transform: sc
10    -webkit-transform
11    -moz-transform: s
12    -o-transform: sca
13    transform: scale(
14 }
15
```


CSS3 Transform

- Skew

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <link rel="stylesheet" type="text/css" href="style.css"
7   />
8 </head>
9 <body>
10  <div class="demo1"></div>
11  <div class="demo2"></div>
12 </body>
13 </html>
```

```
1 .demo1,.demo2{
2   width: 300px;
3   height: 200px;
4   background-color: red;
5   margin: 100px;
6   float: left;
7 }
8 .demo1{
9   -ms-transform: skewX(40deg);
10  -webkit-transform: skewX(40deg);
11  -moz-transform: skewX(40deg);
12  -o-transform: skewX(40deg);
13  transform: skewX(40deg);
14 }
15 .demo2{
16   -ms-transform: skewX(-40deg);
17   -webkit-transform: skewX(-40deg);
18   -moz-transform: skewX(-40deg);
19   -o-transform: skewX(-40deg);
20   transform: skewX(-40deg);
21 }
22
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <link rel="stylesheet" type="text/css" href="style.css"
7   />
8 </head>
9 <body>
10  <div class="demo1"></div>
11  <div class="demo2"></div>
12 </body>
13 </html>
```

```
1 .demo1,.demo2{
2   width: 300px;
3   height: 200px;
4   background-color: red;
5   margin: 100px;
6   float: left;
7 }
8 .demo1{
9   -ms-transform: skewY(40deg);
10  -webkit-transform: skewY(40deg);
11  -moz-transform: skewY(40deg);
12  -o-transform: skewY(40deg);
13  transform: skewY(40deg);
14 }
15 .demo2{
16   -ms-transform: skewY(-40deg);
17   -webkit-transform: skewY(-40deg);
18   -moz-transform: skewY(-40deg);
19   -o-transform: skewY(-40deg);
20   transform: skewY(-40deg);
21 }
22
```

CSS3 Transform

- 3D Rotate

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <link rel="stylesheet" type="text/css" href="style.css"
  />
7 </head>
8 <body>
9   <div class="demo1">abc</div>
10  <div class="demo2"></div>
11  <div class="demo3"></div>
12 </body>
13 </html>
```

```
1 .demo1,.demo2,.demo3{
2   width: 300px;
3   height: 200px;
4   background-color: red;
5   margin: 100px;
6   transition:all 2s;
7 }
8 .demo1:hover{
9   -ms-transform: rotateX(360deg);
10  -webkit-transform: rotateX(360deg);
11  transform: rotateX(360deg);
12 }
13 .demo2:hover{
14   -ms-transform: rotateY(360deg);
15   -webkit-transform: rotateY(360deg);
16   transform: rotateY(360deg);
17 }
18 .demo3:hover{
19   -ms-transform: rotateZ(360deg);
20   -webkit-transform: rotateZ(360deg);
21   transform: rotateZ(360deg);
22 }
```



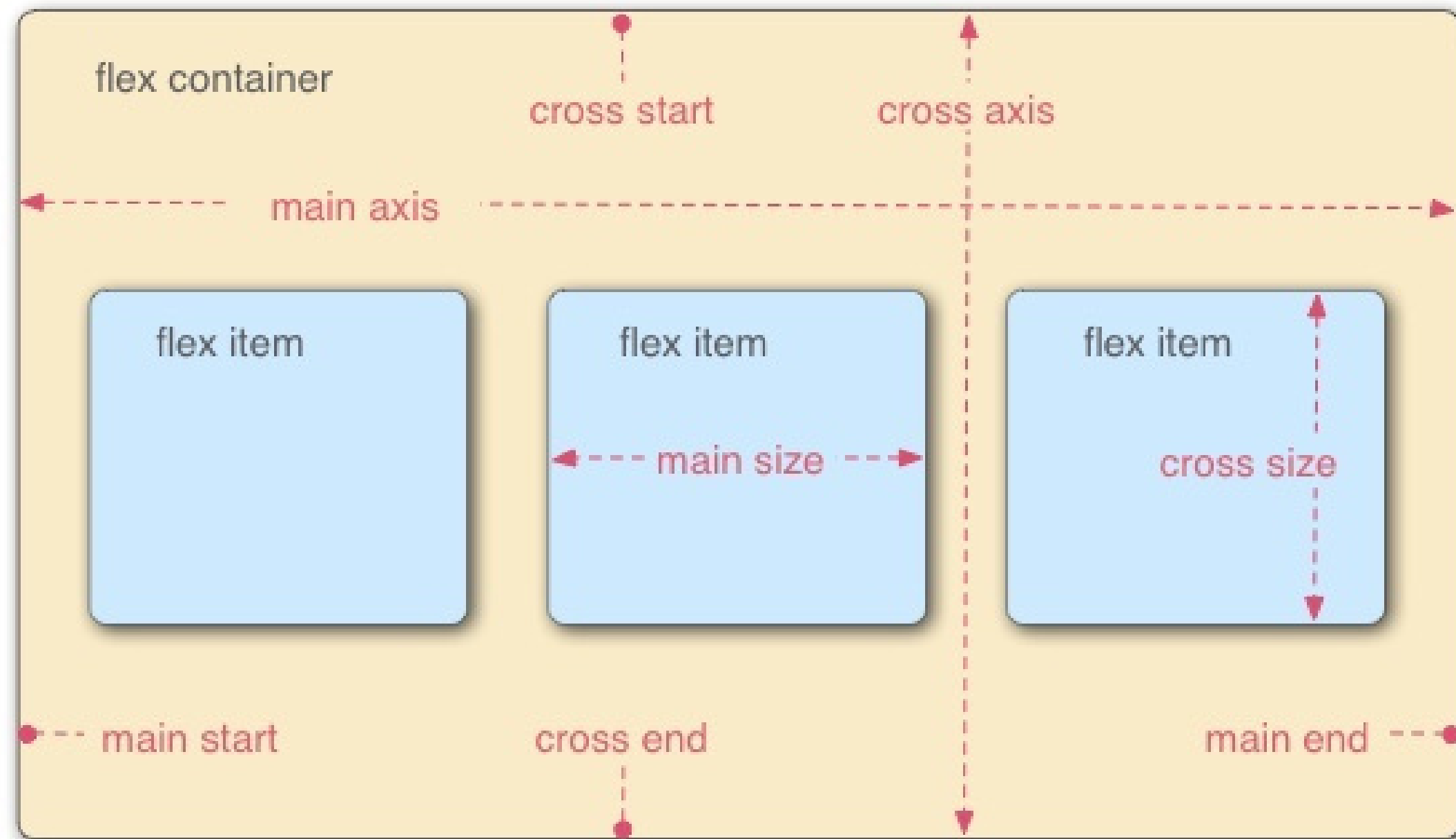
Flexbox



Flexbox là gì

- Thay thế cho thuộc tính float và hạn chế vấn đề của float như clear float
- Tự cân đối kích thước các phần tử trên mọi thiết bị
- Chỉ nên dùng để chia bố cục trong phạm vi nhỏ

Các thành phần trong flexbox

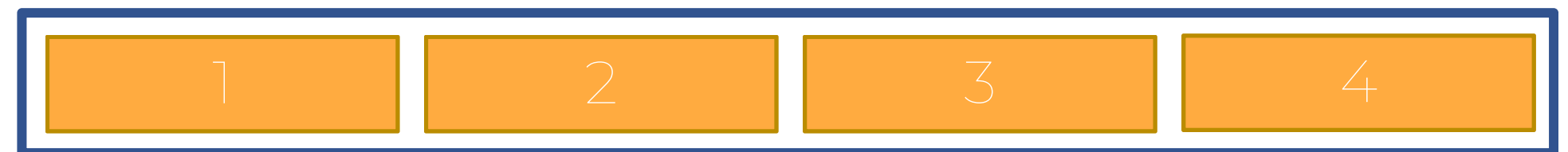
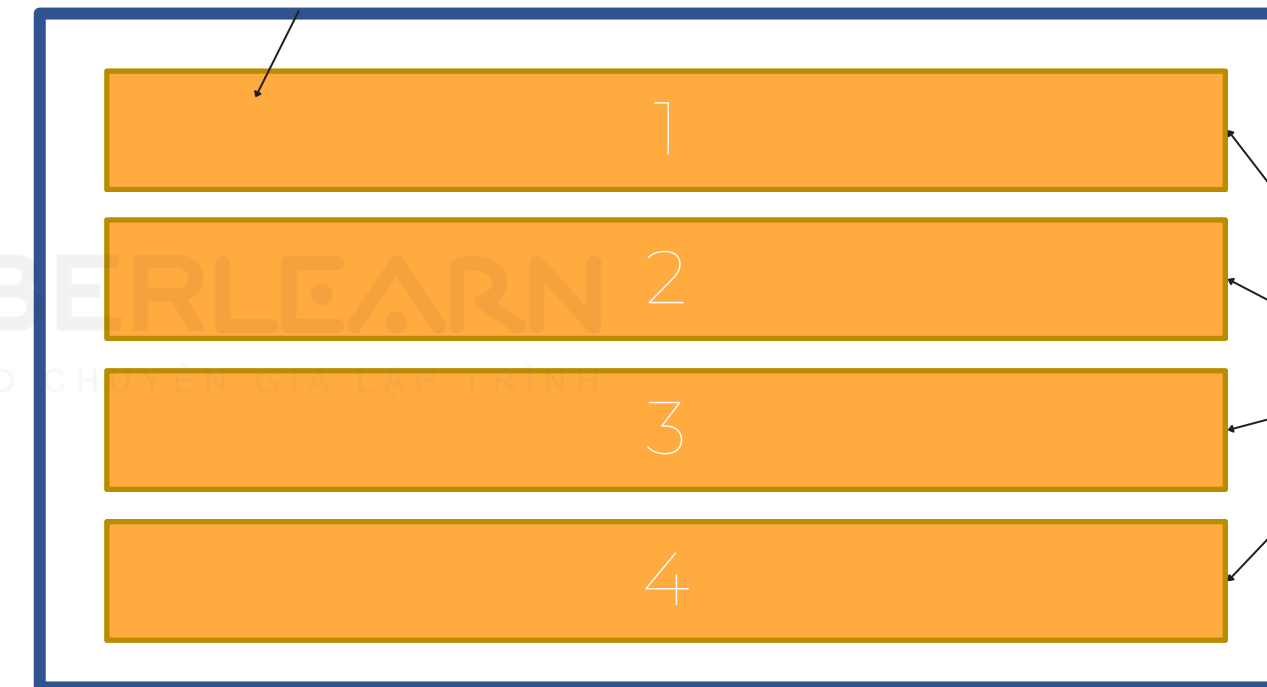


Các thuộc tính của flexbox

- display:
flex

```
.flex-container{  
  display: flex;  
}
```

**Flex-
container**



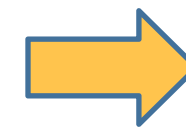
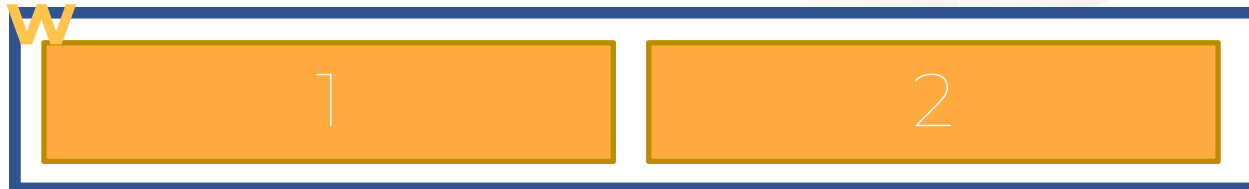
Các thuộc tính của flexbox

- flex-direction:

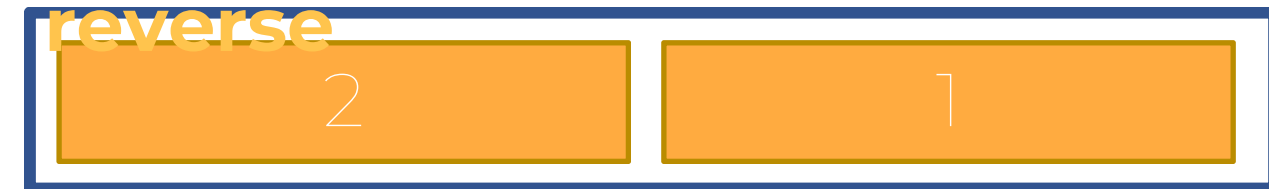
Dùng để đổi hướng sắp xếp của các phần tử

```
.flex-container{  
  flex-direction: row | column | row-reverse | column-reverse;  
}
```

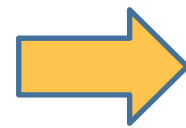
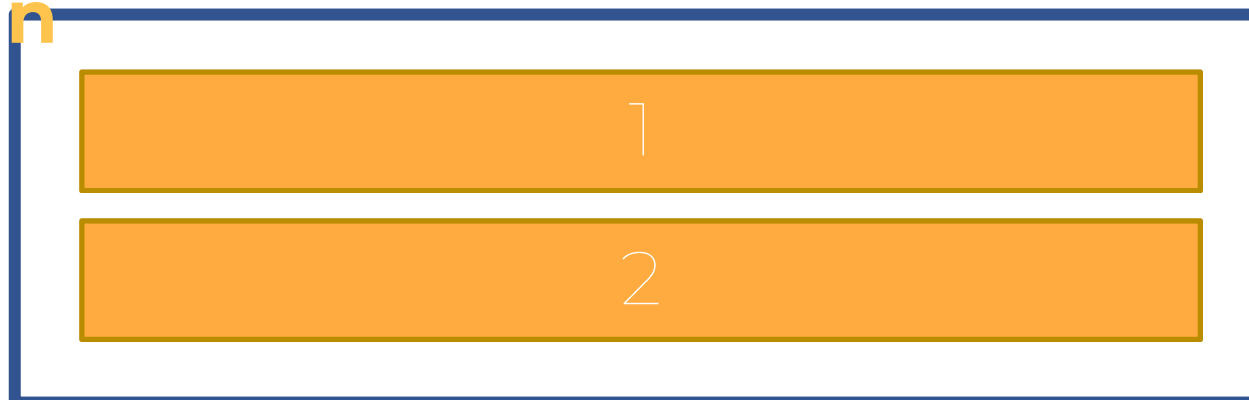
row
w



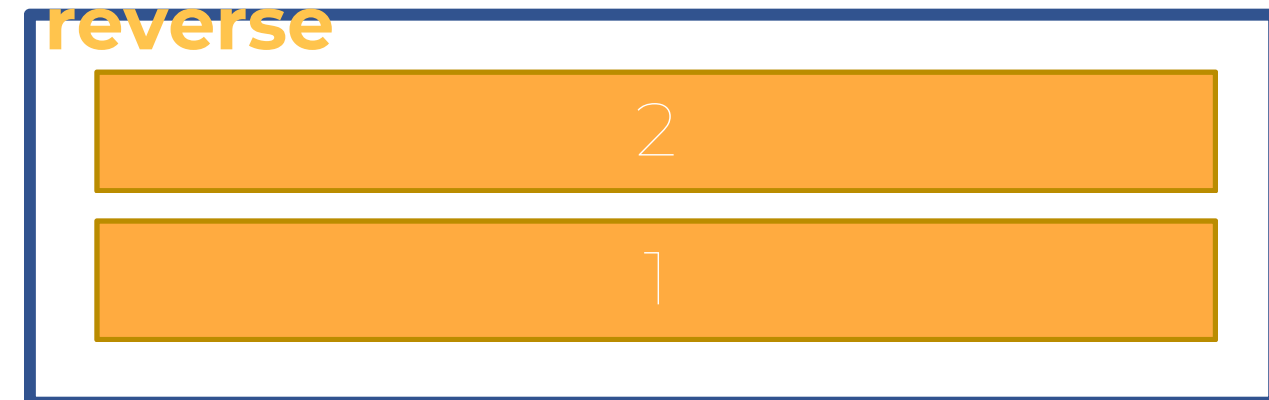
row-
reverse



column
n



column-
reverse



Các thuộc tính của flexbox

- flex-wrap :

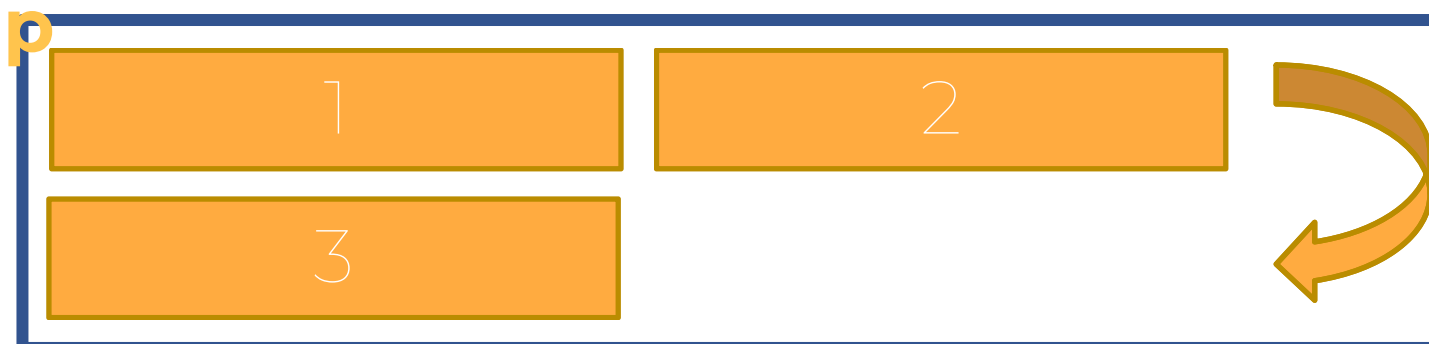
Cho phép các phần tử được xuống hàng khi không đủ chỗ chứa

```
.flex-container{  
  flex-wrap: wrap | nowrap | wrap-reverse;  
}
```

nowrap



wrap



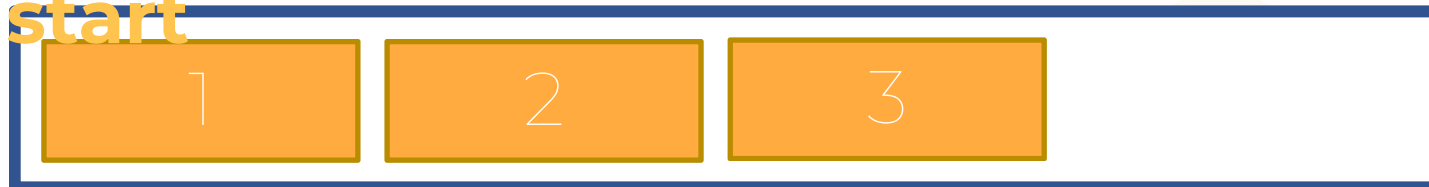
Các thuộc tính của flexbox

- justify-content

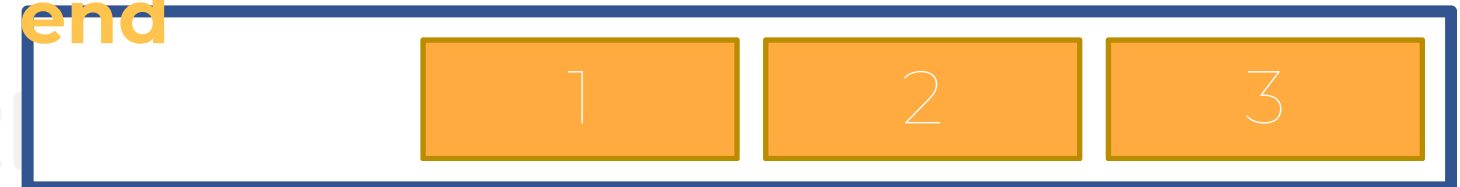
Canh chỉnh vị trí phần tử theo chiều ngang

```
.flex-container{  
  justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;  
}
```

Flex-
start



Flex-
end



cent
er



Space-
around



Space-
between



Space-
evenly



Các thuộc tính của flexbox

- Align-item

Canh chỉnh vị trí phần tử theo chiều dọc

```
.flex-container{  
  ... align-items: flex-start | flex-end | center;  
}
```

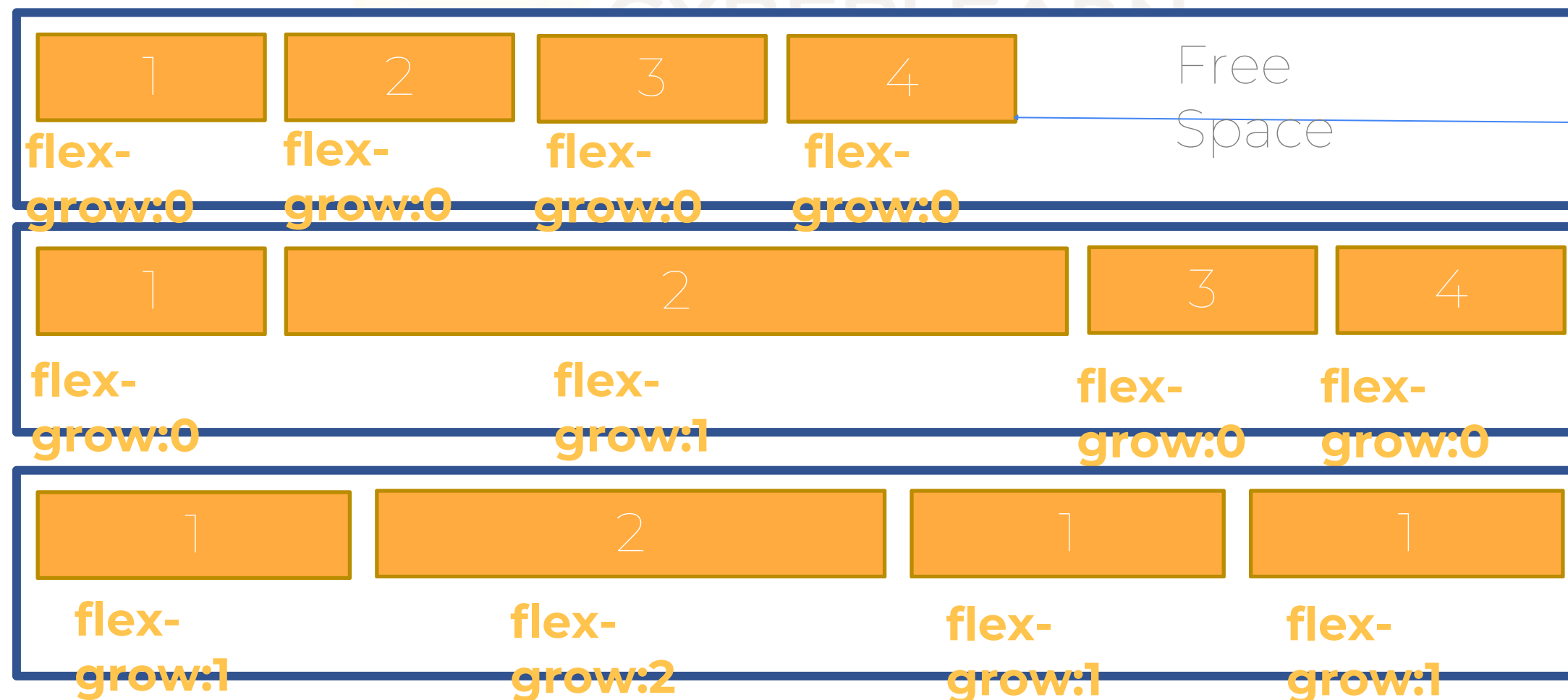
**Flex-
start**



**Flex-
end**

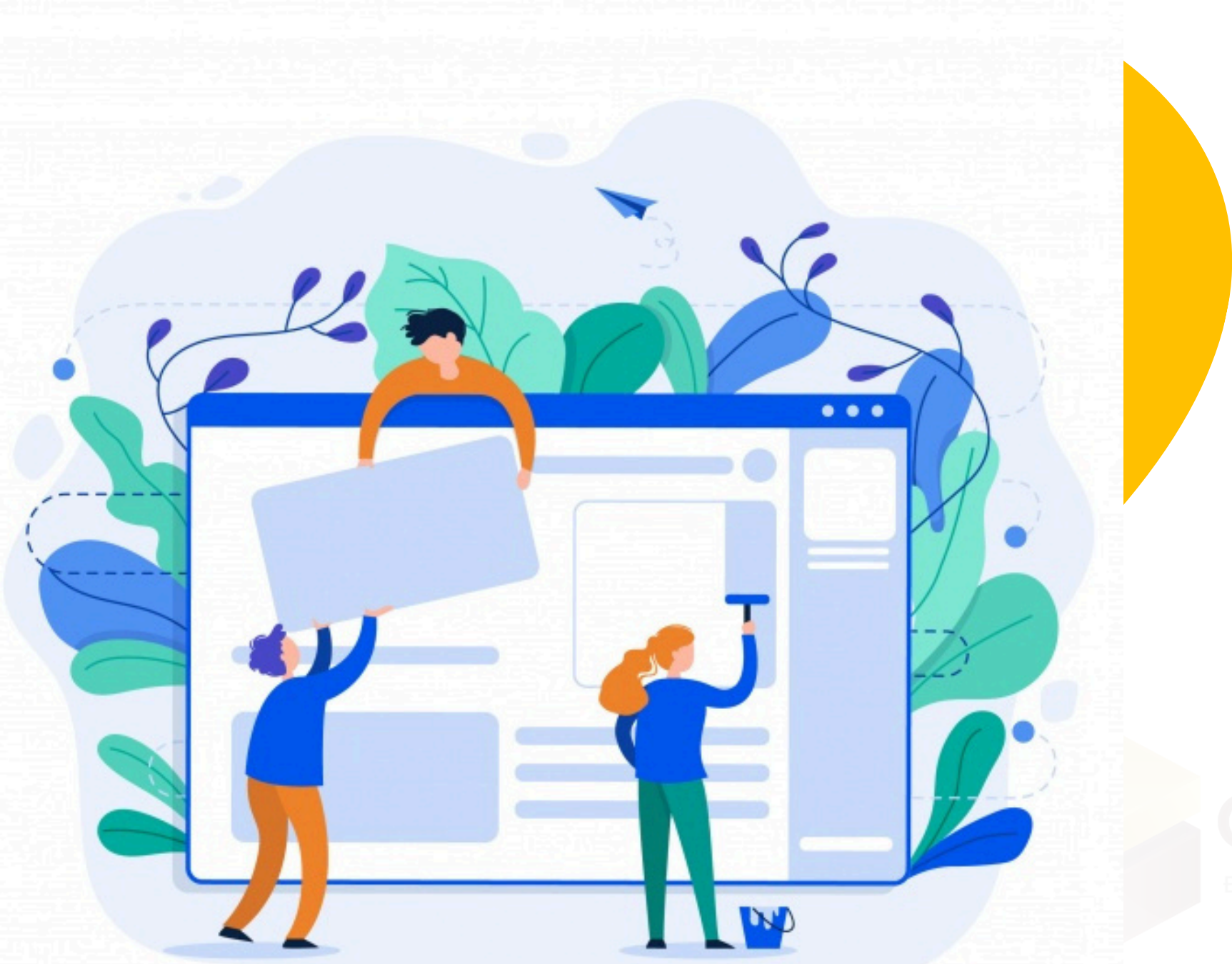
Các thuộc tính của flexbox

- flex-grow:
 - Item được khai báo thuộc tính này sẽ được chiếm phần khoảng trống còn lại của flex.
 - Mặc định: mỗi item sẽ có flex-grow là 0
 - Nếu 1 item có flex-grow:2, các item còn lại là 1 thì phần trống thêm vào cho item đó sẽ gấp 2 lần so với các item khác





Grid



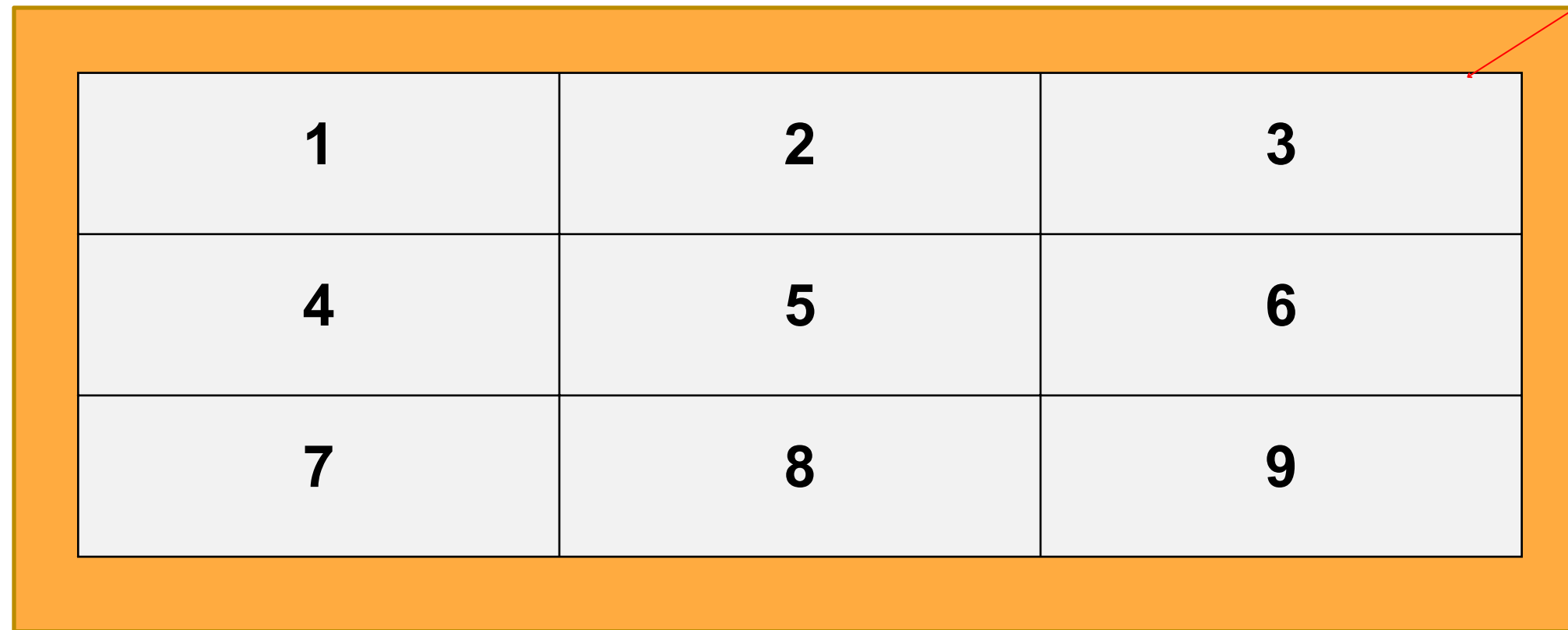
Grid là gì?

- Cung cấp một hệ thống lưới cơ bản
- Bao gồm các cột và các hàng
- Dàn layout dễ hơn mà không phải sử dụng float hay position.

Các thành phần trong Grid

- Gồm 2 thành phần chính :
 - Grid Container
 - Grid Item
- Muốn sử dụng css Grid phải có các thuộc tính :
 - Display: grid
 - Grid-template-columns: chia cột cho phần tử

Item



1	2	3
4	5	6
7	8	9

Container

Các thuộc tính của Grid

- Grid-template-columns
 - Dùng để chia cột cho phần tử
 - Cú pháp:
 - Grid-template-columns:[col1] [col2] [col3]...;
 - Grid-template-columns: repeat(số cột, kích thước);
 - Đơn vị kích thước ưu tiên dùng “fr”
 - Ví dụ: grid-template-columns: 1fr 1fr 2fr
=> chia phần tử ra thành 4 phần bằng nhau, cột 1 và 2 chiếm một phần, cột 3 chiếm 2 phần

Column	Column	Column

Các thuộc tính của Grid

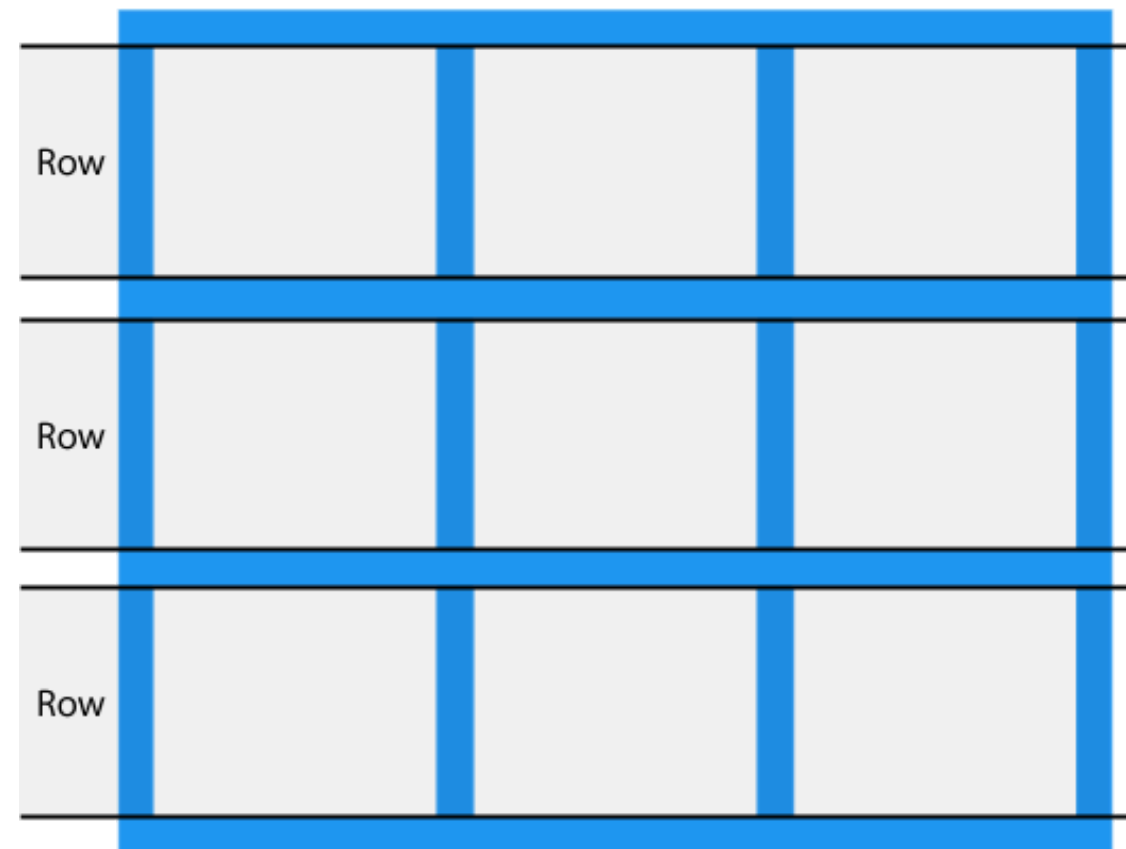
- Grid-template-columns

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 10px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}
```

1	2	3
4	5	6

Các thuộc tính của Grid

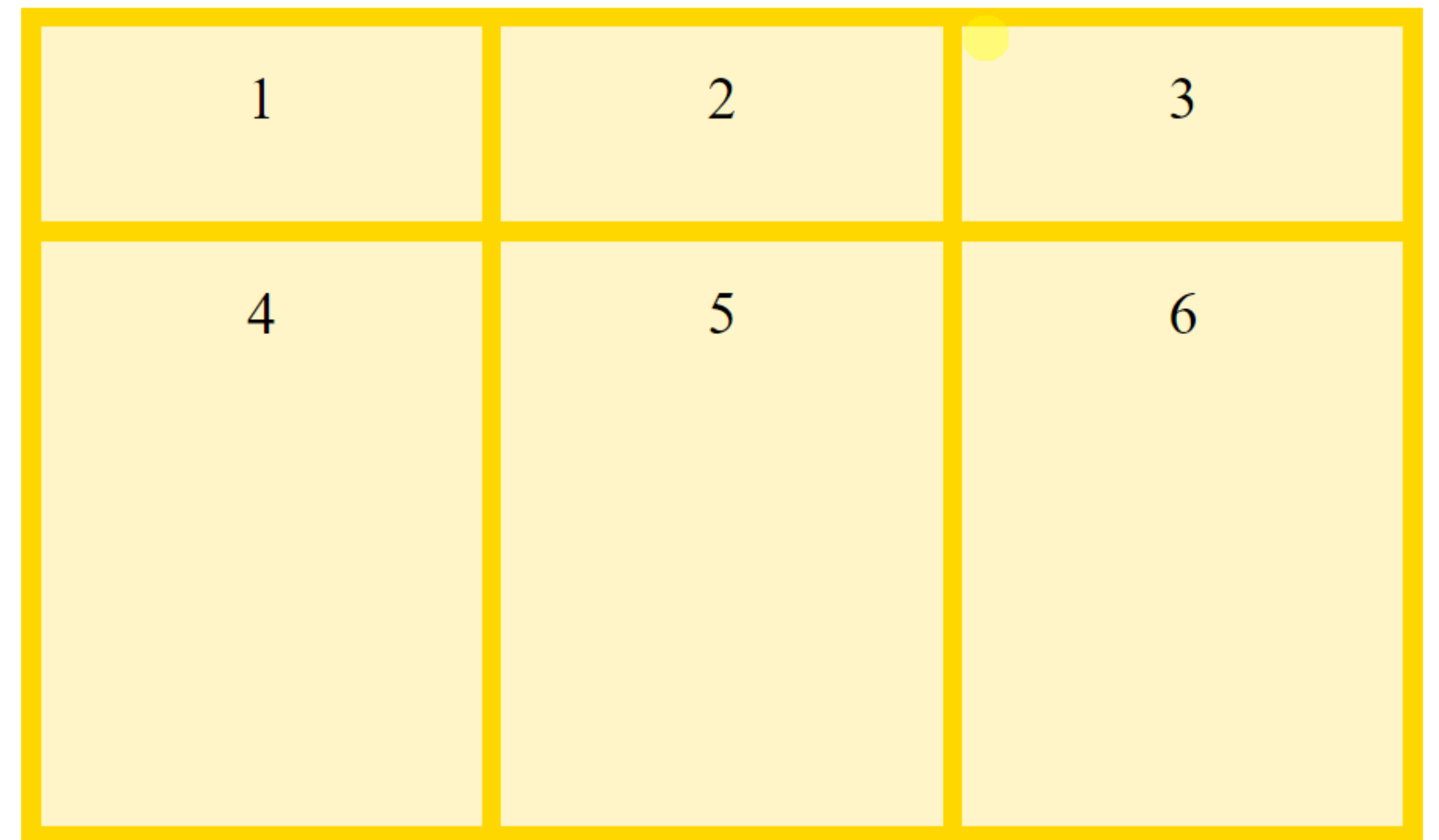
- Grid-template-rows
 - Cú pháp :
 - Grid-template-rows:[row1] [row2] [row3]...;
 - Chỉ nên xét khi muốn chỉnh height cho các rows.



Các thuộc tính của Grid

- Grid-template-rows

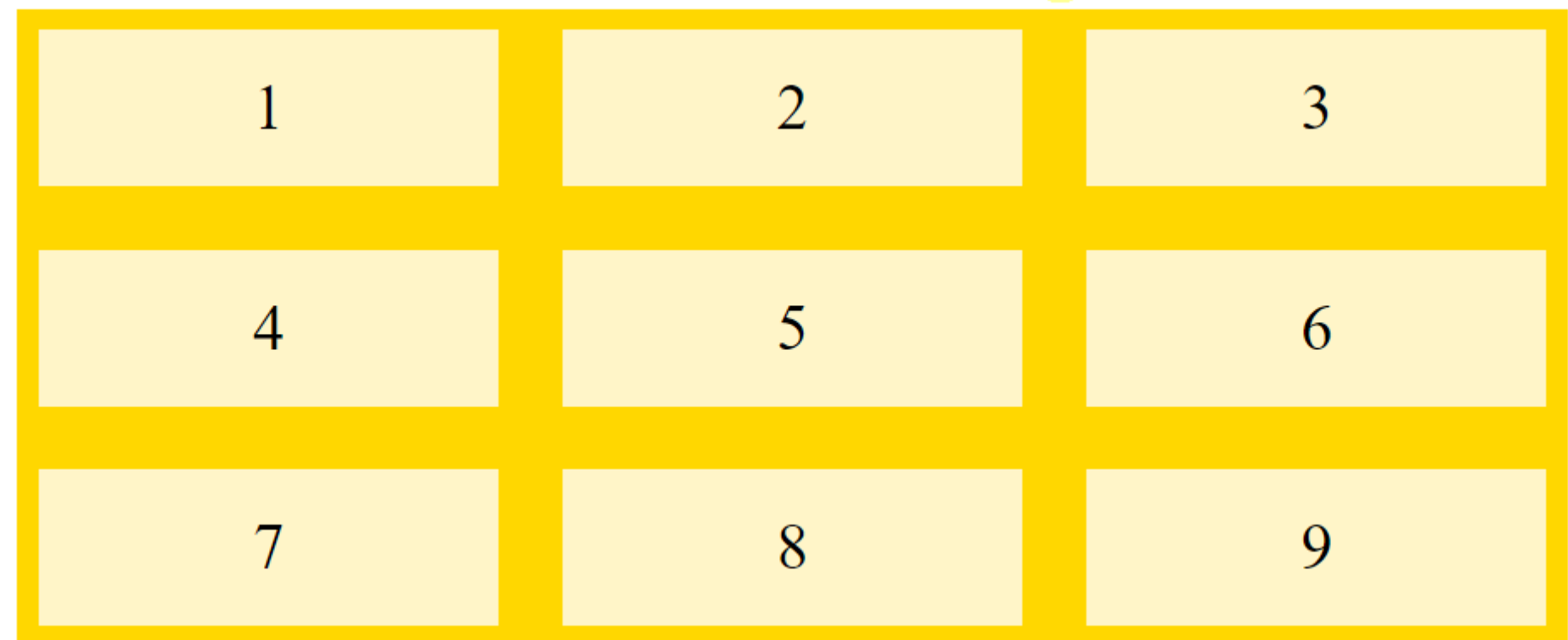
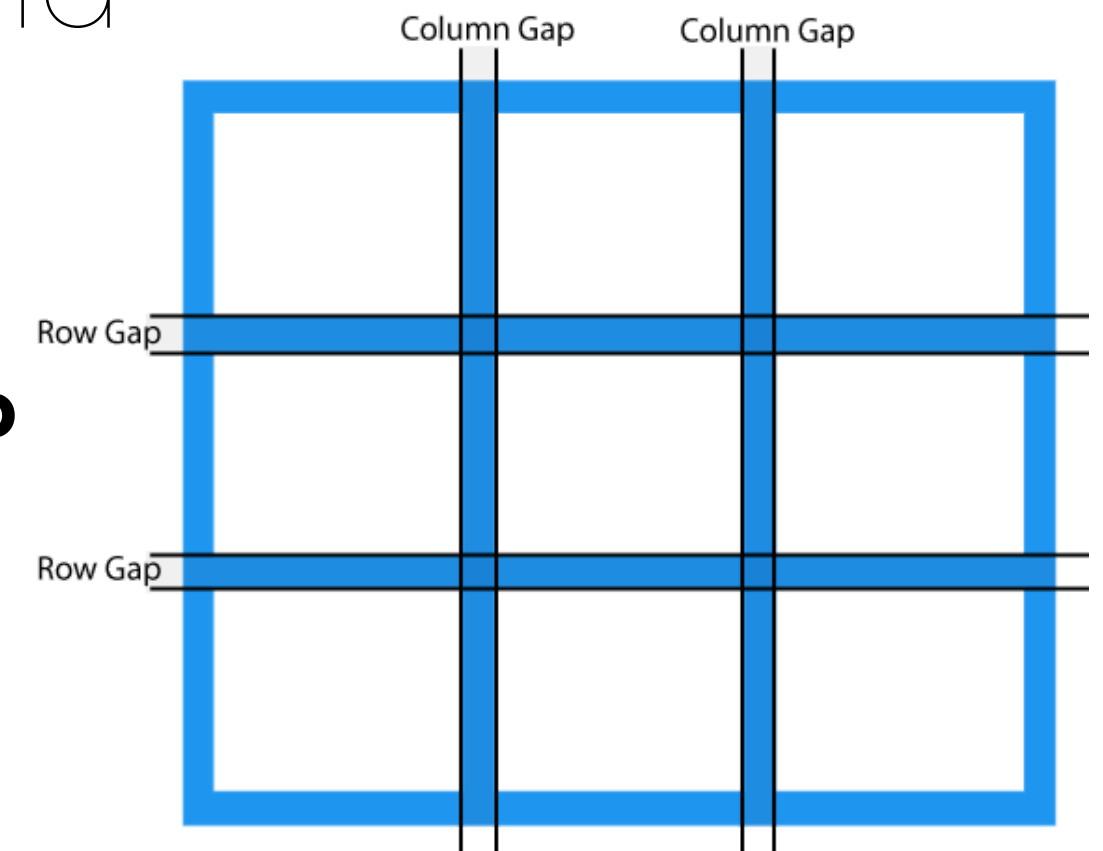
```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-template-rows: 100px 300px;  
  grid-gap: 10px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}
```



Các thuộc tính của Grid

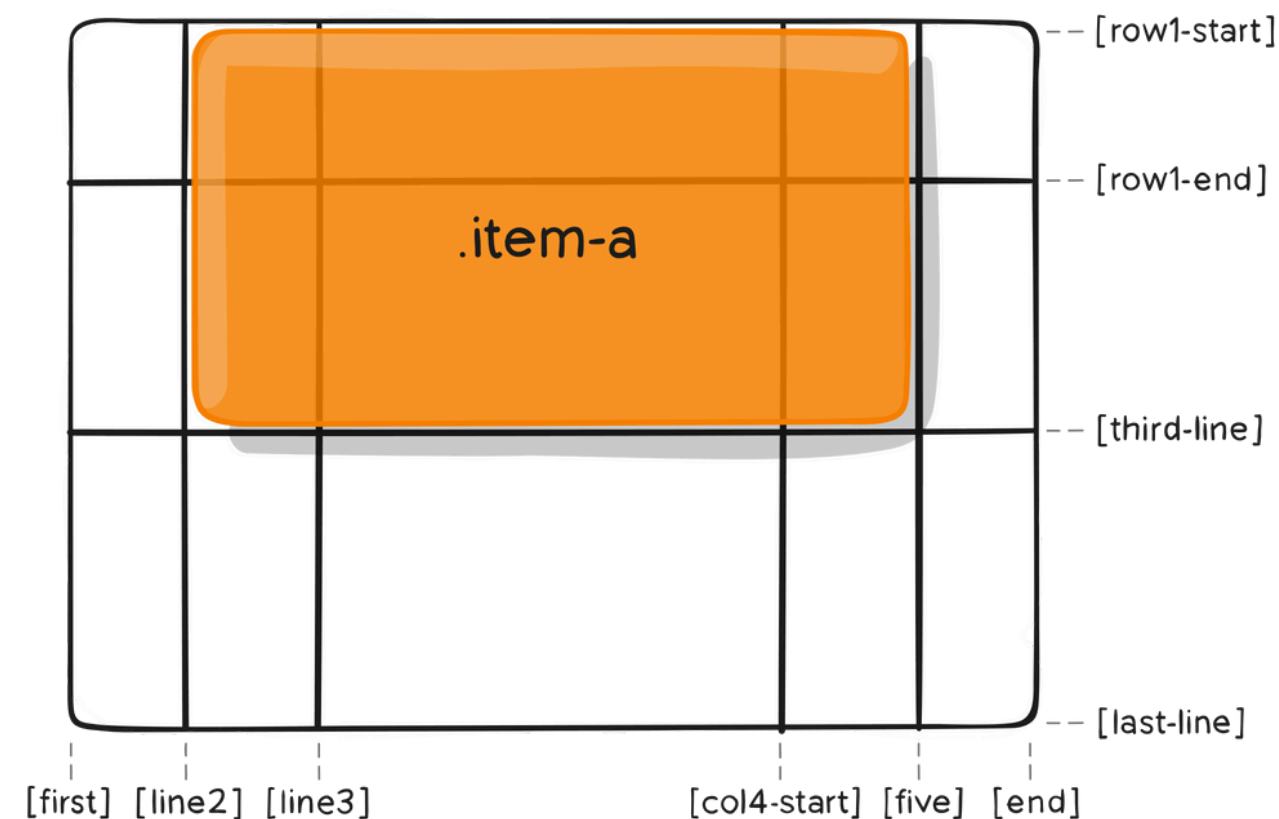
- Grid Gaps
 - Là khoảng cách giữa các cột và các hàng.
 - Cú pháp: **Grid-column-gap | grid-row-gap | grid-gap : [khoảng cách];**

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 30px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}
```



Các thuộc tính của Grid

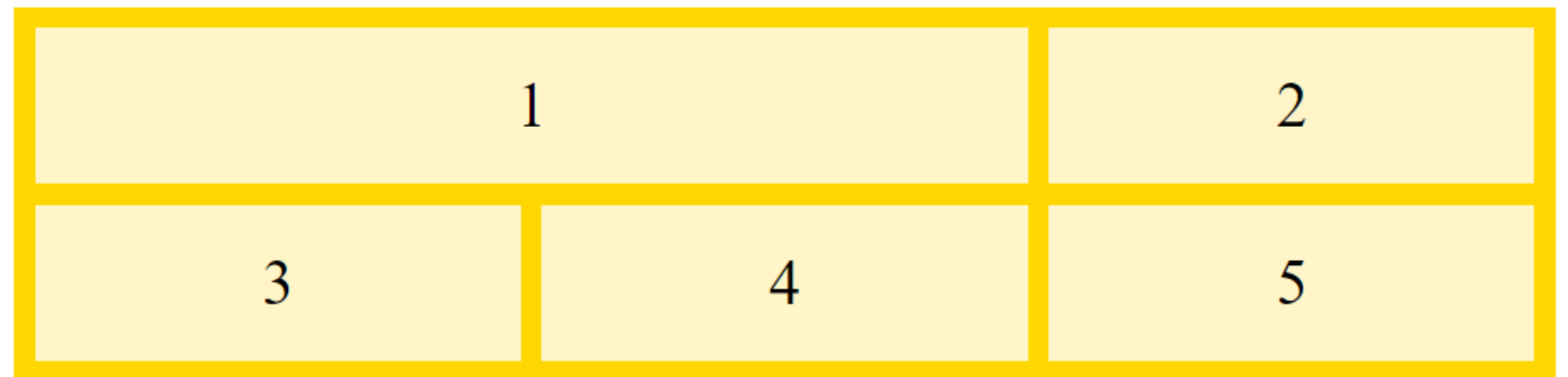
- Grid-column & Grid-row:
 - Dùng để gom cột hoặc hàng tương tự như table
 - Thêm ở item con cần gom
 - Cần xác định **line** bắt đầu và kết thúc của item đó.
 - Ví dụ: Item-a chiếm từ cột 2 đến cột 4 và hàng 1 đến hàng 2 nên sẽ phủ từ [line2] đến [five] và từ [row1-start] đến [third-line]



Các thuộc tính của Grid

- Grid-column:
 - Cú pháp: **grid-column: n / m;**
 - Trong đó:
 - n là vị trí line bắt đầu của cột đầu tiên mà item con này chiếm
 - m là vị trí line kết thúc của cột cuối cùng mà item con này chiếm

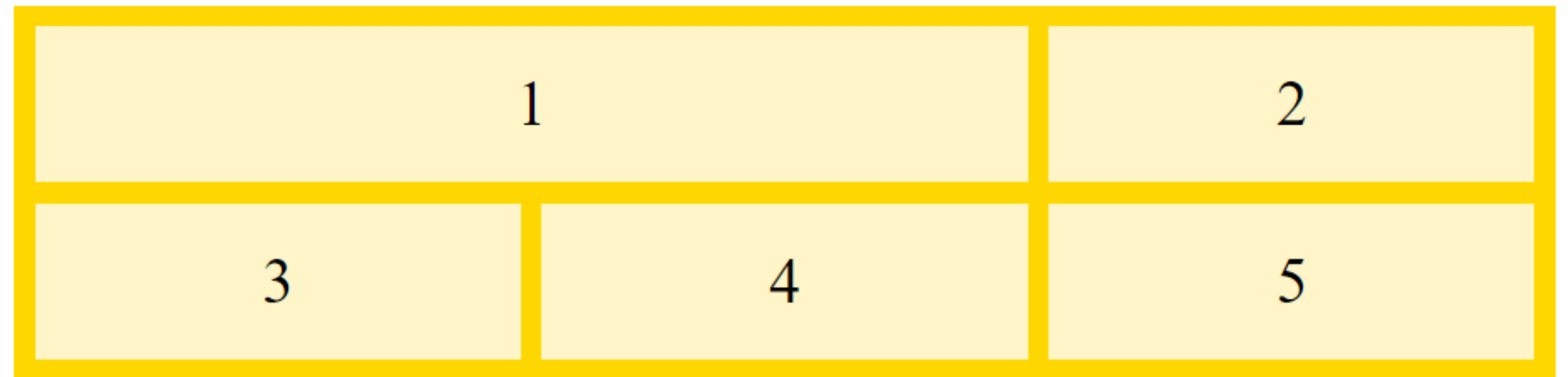
```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 10px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
  
.item1 {  
  grid-column: 1 / 3;  
}
```



Các thuộc tính của Grid

- Grid-column:
 - Cú pháp: **grid-column: n / span m**
 - Trong đó:
 - n là vị trí line bắt đầu của cột đầu tiên mà item con này chiếm
 - m là số cột ta muốn phần tử này chiếm

```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 10px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
  
.item1 {  
  grid-column: 1 / span 2;  
}
```



Các thuộc tính của Grid

- Grid-row:
 - Cú pháp gần giống với grid-column
 - **grid-row: n / m**
 - **grid-row: n / span row**
 - Trong đó:
 - n là vị trí line bắt đầu của hàng đầu tiên mà item con này chiếm
 - m là vị trí line kết thúc của hàng cuối cùng mà item con này chiếm
 - row : số hàng mà item con này chiếm



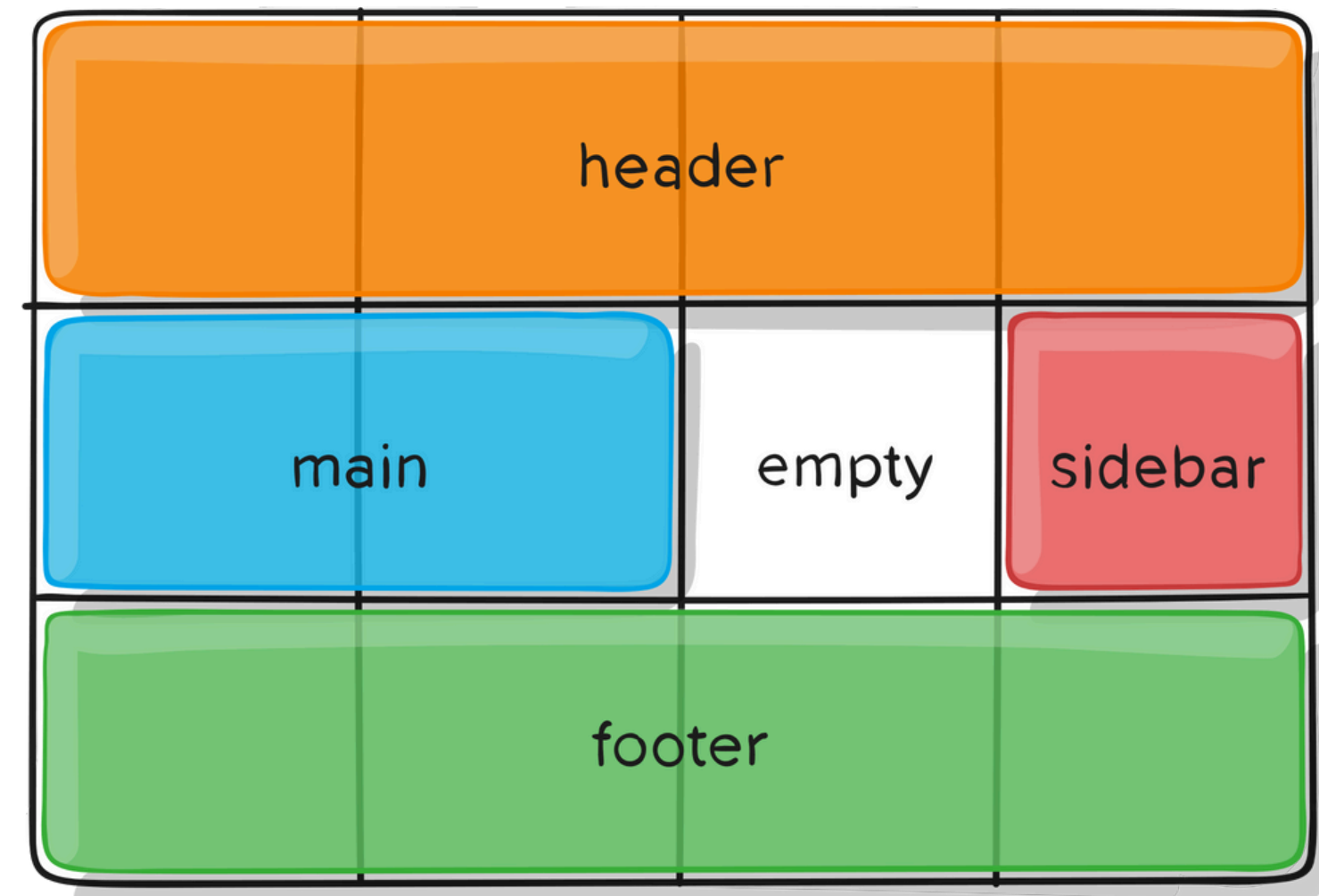
```
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  grid-gap: 10px;  
  background-color: gold;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
  
.item1 {  
  grid-row: 1 / span 2;  
}
```

Các thuộc tính của Grid

- Naming Grid Item:

Ta có thể đặt tên cho các grid items để sắp xếp vị trí trong trang web.

```
.item-a {  
  grid-area: header;  
}  
.item-b {  
  grid-area: main;  
}  
.item-c {  
  grid-area: sidebar;  
}  
.item-d {  
  grid-area: footer;  
}  
  
.container {  
  display: grid;  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}
```



So sánh Float, Flexbox, Grid

	Float	Flexbox	Grid
Phiên bản	CSS	CSS3	CSS3
Khai báo	con	• Thêm ở thẻ cha	• Số lượng column
Hỗ Trợ	• Xét width cho • Chỉ nên dùng khi flex và grid	nam trên 1 nang • Hỗ trợ cho các	• Số lượng row (nếu năng và cột • Hỗ trợ cho các



So sánh Float, Flexbox, Grid

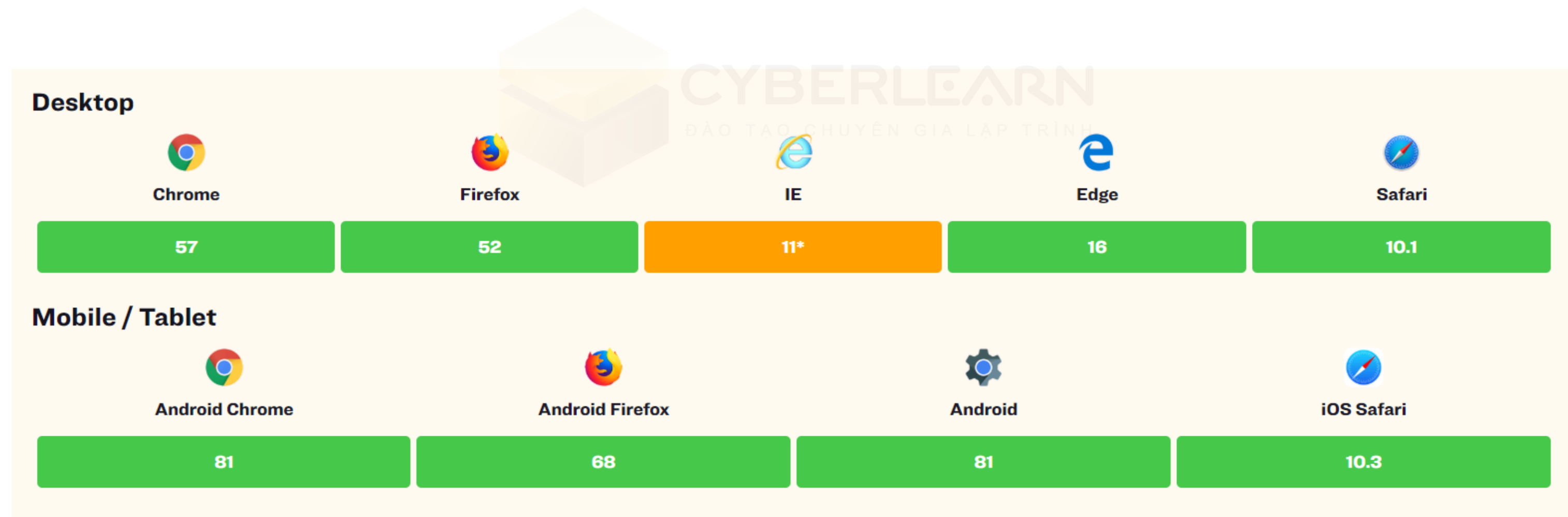
- Browser Support cho flex:
 - (new): khai báo display: flex;
 - (tweener): khai báo display: flexbox;
 - (old): khai báo display: box;



Chrome	Safari	Firefox	Opera	IE	Edge	Android	iOS
20- (old) 21+ (new)	3.1+ (old) 6.1+ (new)	2-21 (old) 22+ (new)	12.1+ (new)	10 (tweener) 11+ (new)	17+ (new)	2.1+ (old) 4.4+ (new)	3.2+ (old) 7.1+ (new)

So sánh Float, Flexbox, Grid

- Browser Support cho grid:
Nếu như trình duyệt không hỗ trợ grid thì dùng flexbox thay thế, trường hợp cả flex và grid đều không được thì dùng float





CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Animation

CSS3 Animation

- Như ta đã thấy, transform và transition thường kết hợp với 1 sự kiện xảy ra
- Với animation, ta có set up hiệu ứng theo ý muốn và có thể mặc định chạy khi load trang web

CSS3 Animation

- Tên hiệu ứng được đặt thoải mái.
- Phải có duration.
- Có thể chỉnh theo from...to hoặc theo %

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Document</title>
6   <link rel="stylesheet" type="text/css" href="
7     style.css" />
8 </head>
9 <body>
10   <div class="demo1">abc</div>
11 </body>
12 </html>
```

```
1
2 .demo1{
3   width: 200px;
4   height: 200px;
5   background-color: red;
6   -webkit-animation-name: hieudeptrai;
7   -webkit-animation-duration: 2s;
8   animation-name: hieudeptrai;
9   animation-duration: 2s;
10 }
11 @keyframes hieudeptrai {
12   from{
13     width: 200px;
14   }
15   to{
16     width: 600px;
17   }
18 }
19
```

```
.demo1{
  width: 200px;
  height: 200px;
  background-color: red;
  -webkit-animation-name: hieudeptrai;
  -webkit-animation-duration: 2;
  animation-name: hieudeptrai;
  animation-duration: 4s;
}
@keyframes hieudeptrai {
  0%{
    width: 200px;
  }
  50%{
    width: 600px;
  }
  100%{
    width: 200px;
  }
}
```

CSS3 Animation

- Animation delay
 - Khoảng thời gian delay trước khi thực hiện hiệu ứng.
 - Cú pháp:
 - Animation-delay: [Time] ;
 - Ví dụ : Animation-delay : 1s;

CSS3 Animation

- Animation iteration-count
 - Số lần lặp của hiệu ứng.
 - Cú pháp: Animation-iteration-count: [Times] ;
 - Ví dụ :
 - Animation-iteration-count: 3;
 - Animation-iteration-count: infinite;

CSS3 Animation

- Animation timing-function
 - Xác định tốc độ của hiệu ứng
 - Cú pháp: Animation-timing-function: [Types] ;
 - Ví dụ :
 - Animation-timing-function : ease;
 - Animation-timing-function : linear;
 - Animation-timing-function : ease-in;
 - Animation-timing-function : ease-out;
 - Animation-timing-function : ease-in-out;

CSS3 Animation

- Animation direction
 - Hướng chuyển động của animation
 - Cú pháp: animation-direction: [Types] ;
 - Ví dụ :
 - animation-direction : normal;
 - animation-direction : reverse;
 - animation-direction : alternate;
 - animation-direction : alternate-reverse;

Loại	Mô tả
normal	Giá trị mặc định, chuyển động theo hướng được định nghĩa trong keyframe
reverse	Chuyển động theo chiều ngược với hướng được định nghĩa trong keyframe
alternate	Chuyển động theo hướng cùng chiều keyframe sau đó đi chiều ngược lại và dừng ở trạng thái đầu tiên
alternate-reverse	cùng chiều rồi dừng lại ở trạng thái cuối cùng của keyframe

CSS3 Animation

- Animation fill-mode
 - Định nghĩa css của thẻ trước hoặc sau khi animation chạy hoặc cả hai
 - Cú pháp: animation-fill-mode: [Types] ;
 - Ví dụ :
 - animation-fill-mode : none;
 - animation-fill-mode : forwards;
 - animation-fill-mode : backwards;
 - animation-fill-mode : both;

Loại	Mô tả
none	Giá trị mặc định, chuyển động theo css được định nghĩa trong keyframe
forwards	Thẻ sẽ duy trì css của keyframe cuối cùng sau khi animation kết thúc
backwards	Thẻ sẽ lấy css của keyframe đầu tiên sau đó mới bắt đầu animation
both	animation và duy trì css của keyframe cuối cùng sau khi animation kết thúc